UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Preliminary version of FINDER,

a Pascal program for locating mineral deposits with spatial information

by D. A. Singer

Open-File Report
85-590

This report is preliminary and has not
been reviewed for conformity with
U.S. Geological Survey editorial standards.

Menlo Park, California
1985

Preliminary version of FINDER,

a Pascal program for locating mineral deposits with spatial information

by D. A. Singer

INTRODUCTION

The fundamental problem in exploration is to discriminate responses related to mineralization (hopefully ore deposits) from responses related to barren areas. Knowledge of how measured variables behave in and near the type of mineralization sought and in barren areas is required in order to discriminate the populations using samples. Although differences between barren and mineralized samples may be striking, more commonly there is an overlap in the frequencies of the two groups so that some samples could have come from either population. For example, a volcanic rock sample having 0.05 percent mercury might be a high background sample or it might be from the altered rocks around an epithermal precious metal deposit. One way to determine which population the sample is from is to consider the responses of nearby samples; if many of the adjacent samples contain less than 0.01 percent mercury then the sample is probably from the barren population. How close the samples should be and how many should be below some limit requires knowledge of the characteristics of the variable near and away from the deposit type and a means to integrate the responses and their spatial patterns.

FINDER, a recently developed computer program employing principals of artificial intelligence to aid in selection of target areas on the basis of one or more geologic, geochemical, and geophysical variables, and multiple observations, can be used to integrate the responses with the spatial patterns. The method used is a combination of the area of influence procedure (Singer and Drew, 1976) and Bayesian statistics (Raiffa, 1968). A similar idea relying on simulation and Bayesian statistics was presented by Rehder and van den Boom (1983) for circular and elliptical-shaped geochemical variables. FINDER is the first published program utilizing the more efficient and powerful area of influence method (Singer, 1976) and Bayesian statistics. In addition to circular and elliptical target shapes, FINDER allows annular and annular elliptical-shaped targets, preferred orientations, and provides an estimate of the number of deposits.

MATHEMATICS OF THE PROBLEM

In a well-studied control area there is a circular shaped mineralized target that has a radius r (Figure 1A). Figure 1B shows hypothetical frequency distributions of samples of a variable taken within the radius r and samples taken outside the radius r. A sample with a value of 10 would be clearly identified as belonging to the barren population, but samples with values between 15 to 40 could have come from either population. If there are only two possibilities, barren or mineralized, then the probability that a group of samples came from the mineralized population can be calculated by means of Bayes Rule (Raiffa, 1968) as follows:

$$P(\text{mineralized}:X_1,\ldots,Xn) = \frac{Pm \prod_{i=1}^{n} fm(Xi)}{Pm \prod_{i=1}^{n} fm(Xi) + [1-Pm] \prod_{i=1}^{n} fb(Xi)} \quad (1)$$

where;

$P(\text{mineralized}:X_1,\ldots,Xn)$ is the revised probability of mineralization, given the observations $X1,\ldots,Xn$;

Pm is the prior probability of mineralization, typically a small number like 0.005 is used initially;

$\prod fm(Xi)$ is the product of the probabilities of each sample or variable being from the mineralized population;

$\prod fb(Xi)$ is the product of the probabilities of each sample or variable being from the barren population; and,

n is the number of samples or variables.

The probability of each sample being from the mineralized population, $fm(Xi)$, can be estimated from the observed relative frequency or it can be calculated from a probability density function. In FINDER, the normal probability density function is used to estimate the probabilities of samples belonging to both the barren, $fb(Xi)$, and the mineralized, $fm(Xi)$, populations. In general:

$$P(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp[-1/2\left(\frac{x-\mu}{\sigma}\right)] \quad (2)$$

where;

$P(x,\mu,\sigma)$ = probability fm or fb,

x = observation,

μ = population mean, and

σ = population standard deviation.

From the control area, the estimated population mean and standard deviation of the barren and the mean and standard deviation of the mineralized samples are calculated and used with the normal probability density function, equation 2 (Hogg and Craig, 1965) to calculate probabilities used in Bayes Rule (equation 1). Bayes Rule allows the effects of multiple samples and multiple variables to be represented in the estimated probability. It is important to note that if multiple variables are employed, the variables must be independent. Independence can be achieved by the use of discriminant analysis, factor analysis, or linear regression. Thus for properly transformed variables, a mechanism is provided for integrating multiple variables to make a probabilistic statement concerning the likelihood that a particular locality lies within a mineralized or barren area. In order to include the effects of multiple samples in the estimate, it is necessary to consider the spatial arrangement of the samples with respect to the target sought.

The area of influence procedure provides a mechanism of relating the spatial pattern of observations to the spatial pattern of information concerning the presence or absence of a target that has a particular size and shape.

Methods employed in the FINDER program rely upon the geometry of targets and the positions of samples or drill holes. The map plan patterns of many

variables related to mineralized areas can be approximated by ellipses, circles, annuluses, or annular ellipses. The key to understanding the area of influence procedure is the mapping of these target patterns onto the positions of the samples. Thus, in Figure 1A, any sample from within radius r of the center of the circular target should reflect the characteristics of the target whereas any sample beyond radius r should reflect the barren population. If Figure 1A were divided into a large number of small cells each of which represented a sample location, then circles of radius r drawn around every sample within the target would overlap the most over the center of the target. The mapping of the target onto sample locations allows statements to be made about possible centers of targets. In cases where it is certain that a sample came from the target of interest, then the probability is 1.0 that the target is centered somewhere within radius r of the sample. Similarly, where the sample definitely did not come from target of interest, the target's center could not be within radius r of the sample.

For elliptically shaped targets the same logic is applied; points less than the target semiminor axis from a sample can be treated like the circular target. The probabilities of a mineralized area at points greater than the target semimajor axis from the sample are not affected by the sample. For points between the semiminor and semimajor axis lengths away from the sample, the effects of orientation must be considered. The central angle of possible orientations affected (Drew, 1966; Singer and Drew, 1976) is calculated as follows:

$$\theta = 2\tan^{-1} \frac{b}{a}\sqrt{\frac{(a^2 - d^2)}{(d^2 - b^2)}} \qquad \text{for } a > d > b \qquad (3)$$

where, a is the length of the semimajor axis of the target ellipse, b is the length of the semiminor axis of the target, and d is the distance between the sample and the point being considered. Equations 1 and 2 are applied to the possible orientations affected.

In the FINDER program, a fine grid for the study area is defined by the number of points (cells) to be plotted in the X and Y direction. Sequentially, the distance between cell centers and sample locations are compared with the radius of the circular target or the length of the semimajor axis of the target ellipse. For circular targets the cell's probability is calculated directly from Equations 1 and 2. For elliptical targets, Equation 3 is used with the angle between the cell and sample to determine the affected orientations. The product of the probabilities belonging to the mineralized population are placed in the appropriate one degree step position of θ in arrays called DIM and DIB respectively.

Equations 1 and 2 are then applied to each of the one degree steps. The prior probability is used for the unaffected orientations. Finally, the probabilities of each one degree step are summed and divided by the number of degrees possible. Where there is no preferred orientation declared, 180 (degrees) is used, whereas if a preferred orientation is declared, the number of degrees in the restricted range of orientation is employed. Annuluses or annular ellipses are treated in the same manner as circles or ellipses except that the inner ring is given the characteristics of a barren population circle. Because equation 1 requires independent estimates of fm and fb, it is necessary that adjacent samples not be correlated.

The expected number of deposits is calculated by summing the cell probabilities for the cells having the maximum probability within the radius of the largest dimension of the largest variable. Thus the estimate of the number of deposits applies to the largest variable only. This method will provide unbiased estimates except where the preferred orientation option is used and several possible targets are adjacent to each other; the method will underestimate the number of deposits in this rare case.

APPLICATIONS OF FINDER

A few hypothetical examples are presented in order to demonstrate some of the features of FINDER. Common to the examples and FINDER in general is the assumption that only two populations are possible. Thus, if values similar to those within the target could be provided from another process, the probability map for the study area might reflect the other process and not the target of interest. Where only two populations are present, the coded probabilities associated with each cell in the map reflect the probability that the cell is a center of the target of interest. Because the cell size typically is small compared to the target size the probabilities are not independent from cell to cell.

Figure 2 is the output from a test of FINDER in which a set of 30 samples is used to search for the center of a hypothetical elliptical target that is known to have a preferred orientation. Sample locations and the target sought are plotted over the map output. The highest probabilities of 0.8 to 0.9, which are represented by the symbol "8" in Figure 2 are located over the target and cover the center of the target. The relationship between plotted symbols and probabilities is provided in the section "Plotted Frequencies" of Figure 2. Lower probabilites in the upper part of the map represent possible centers given the positions of the two samples in the deposit and the lack of samples in the upper region. The estimate of 1.2 deposits reflects the high values in the target plus scattered cells with probabilities between 0.0 and 0.1.

Another application of FINDER is the simulation of sampling designs to determine possible consequences of various sampling strategies and sample densities. The question of whether the sample density is sufficient to provide coverage of possible targets can be addressed by using sample locations that have high values associated with them; this will produce a map showing coverage. If the means of the mineralized and barren populations are not greatly different (relative to the standard deviations) then a higher sample density would be required to provide an adequate chance of discovering a target.

The ability of FINDER to remove high background values is demonstrated in Figures 3A and 3B. In Figure 3A, 25 randomly placed values from a table of random normal values are contoured; the values of +2 and +3 in the lower right part of Figure 3A might be donsidered an interesting area to explore based on the contoured results. For a circular target with a radius of 15 units however, FINDER suggests (Figure 3B) that there are no areas with high probabilities of belonging to the "deposit" population. The sample density is such that the filter action of FINDER would not have worked for a small target however.

In Figure 4, 40 randomly positioned samples demonstrate the use of two variables. All samples were given random normal background values for variable 2 because no samples were positioned within the radius of variable 2. Samples beyond the ellipse of variable 1 were also given random normal background values, whereas samples within the ellipse were given random normal values with the same mean and standard deviation as the mineralized variable 1. Despite variable 2 not being observed, the center of the actual target is located by the "9's" in Figure 4 and the expected number of targets is 0.99.

For cases where there are missing observations of some variable, FINDER can use information from the other variables reported if the missing observations are given a neutral value such that they cannot change the prior estimate. This achieved by creating dummy observations that are midway in standard deviations between the mineralized and barren means. FINDER can also deal with binary variables such as "ore" and "not ore" by using "0" and "1" for the means of baren and mineralized respectively and adjusting the standard deviations to reflect the associated degree of certainty. Low mean values associated with mineralization such as depletion of an element are treated in the same manner as high mean values in FINDER.

DESCRIPTION OF FINDER INPUT

The program prompts for input line by line. After each response, the user should "hit" return.

Line 1. Any title to identify the run. Up to 80 characters or numbers.

Line 2. Prior probability of target in area considered. Any real number between 0.0 and 1.0 will be accepted. A typical value might be 0.001.

Line 3. Enter an integer for the number of variables (between 1 and 4). Do not count the X and Y coordinates.

Line 4. Are any of the variable patterns elliptical in shape? Enter a "y" (lower case only) if yes or a "n" if there are not elliptical variables. If at least one variable is elliptical in shape than all are treated as ellipses. In cases where shapes are mixed, the circular variable can be entered as an ellipse having a semiminor axis slightly less than the semimajor axis thereby closely approximating a circle.

Repeat the sequence Lines 5 to 7 for each variable

Line 5. Size of variable--if any of the variable patterns are elliptical in shape (Line 4 = "y") then enter the semimajor and semiminor axes. If Line 4 = "n" then enter the radius of the circle associated with the variable. The values can be real (with a decimal) and should be separated by a space for elliptical patterns.

Line 6. Enter the mean and standard deviation of the variable within the ellipse or circle. The values can be real and should be separated by

a space.

Line 7. Enter the mean and standard deviation of the variable outside the ellipse or circle. The values can be real and should be separated by a space.

Line 8. Enter a "y" (lower case) if there is a preferred orientation to the target, otherwise enter "n". (Applies to all variables.)

Line 8a. If the response to Line 8 is not "y" then this line is skipped. Enter the minimum and maximum orientations. The orientations should be in integer form (no decimal) with a space between the numbers. The numbers must be between 1 and 180 (degrees) where 90 (degrees) is east.

Line 9. Enter a "y" (lower case) here if any of the variable patterns are annular in shape, otherwise enter a "n" and go to Line 10.

Line 9a. Skips this line if response in Line 9 is not "y". Enter the number of the variable pattern that is annular in shape-- use the same order used in Lines 5-7.

Line 9b. Skip this line if response to Line 9 is not "y". Enter the radius (real number) of the inner ring of the annular variable identified in Line 9a. The program will automatically generate the mean and standard deviation of the new variable represented by the inner ring of the annulus.

Line 10. Enter the number of points (cells) to be plotted in the X direction and the number of points (cells) to be plotted in the Y direction. Use integers separated by a space. All maps have location 1,1 on the lower left of the map. In the X direction, up to 120 cells can be plotted on an Epson printer and 69 cells on standard printers limited to 80 columns.

Line 11. Enter a printer scale factor-- 0.3 or 0.6 for standard, 1.0 for an Epson printer. An Epson printer will print a proper scale map by dropping every other line. A standard printer will print every 1.666 or 3.333 line depending on whether the Pascal compiler adds a line feed to the end of a packed array of characters. Trial and error is the best way to select the 0.3 or 0.6 scale factor.

Line 12. Enter the number of observations or samples (integer). (Up to 1000).

Line(s)13. One line should be entered for each observation or sample indicated on Line 12. Each line should contain (in order), the X and the Y coordinates of the sample location, and one response for each variable. All values should be real (with decimal) and be separated by a space. The X and Y coordinates should be in commensurate units with the map scaling. Samples can be located outside the map boundaries.

COMPUTER DEPENDENCIES

Although FINDER has been tested on several different computers, there are some constraints that may affect a users selection of an appropriate computer. The most important constraints are the size of the program and the speed of execution. The source code is 651 lines or 22525 bytes long, the compiled code is 28,784 bytes long, and the arrays use 19,396 words; for large problems the array size of "highp, highx, and highy" might need to be increased. The length of the compiled code varies with compiler. These specifications mean that some 8-bit computers are too small for FINDER and some Pascal compilers of 16-bit and perhaps 32-bit machines may not be able to handle FINDER. Use of overlays with the first seven procedures and the function "height" might allow FINDER to be run of some of the smaller computers or compilers. In cases where large data sets and or relatively large elliptical variables are used, FINDER can take hours to run on a 16/32-bit computer; an 8-bit computer seems inappropriate for these problems.

Some versions of Pascal require that input and output devises be identified in the procedures that read the keyboard or write to the screen. If the resolution or aspect ratio of the printed map is not satisfactory, it is possible to write the map or the probabilities for the map to a file that can be used with a plotter. Minor program alterations can be made to adjust for these problems.

SUMMARY

The Pascal program FINDER provides a method of integrating spatial and frequency information such that a map of the probabilities of target (deposit) centers is produced and the number of targets is estimated from the information. FINDER can accept up to four independent variables reflecting geologic, geochemical, or geophysical information. The application of FINDER requires that one or more well-studied targets serve as a control in which the means and standards of each variable are estimated near the target (mineralized population) and away from the target (barren population). Circular, elliptical, and annular shaped target variables can be used and study areas in which targets have a preferred orientation are allowed.

REFERENCES

Drew, L. J., 1966, Grid drilling and its application to the search for petroleum: Unpub. Ph.D. thesis, Pennsylvania State University, 141 p.

Hogg, R. V.,and Craig, A. T., 1965, Introduction to mathematical statistics; Second edition, The Macmillian Co., N.Y., 383 p.

Raiffa, Howard, 1968, Decision analysis; Introductory lectures on choices under uncertainty; Addison-Wesley, Reading Massachusetts, 309 p.

Rehder, S. W., and van den Boom, G. P., 1983, Geochemical characterization of tin granites in northern Thailand; in Howarth, R. J., ed., Statistics and data analysis in geochemical prospecting; Elsevier Scientific Publishing Co., Amsterdam, p. 311-323.

Singer, D. A., 1976, RESIN, a FORTRAN IV program for determining the area of influence of samples or drill holes in resource target search; Computers and Geoscience, Vol. 2, p. 249-260.

Singer, D. A. and Drew, L. J., 1976, The area of influence of an exploratory drill hole: Econ. Geol. Vol. 71, No. 3, p. 642-647.

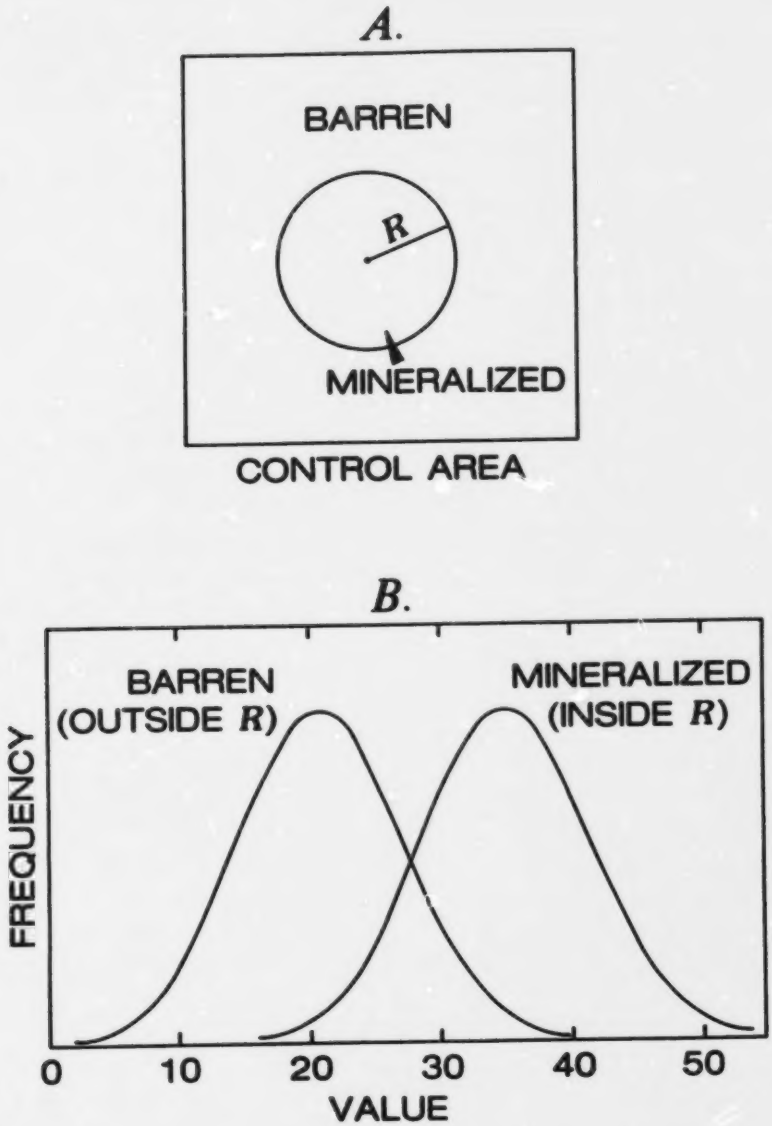
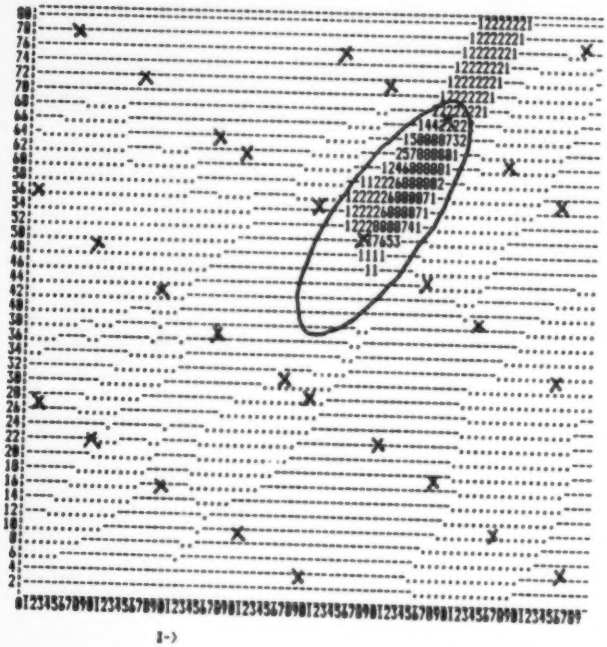


Figure 1-- (A) Well studied control area showing mineralized area within radius R; and (B) Frequency distributions of samples taken within radius R and beyond radius R.

** FINDER PROGRAM **

Test of FINDER with a target having preferred orientation

Prior Probability= .0050



Variable Target Sizes, Associated Means and Stand. Dev. WithIN & OUTside Target

Var	Semimajor axis	Semiminor axis	Mean IN	S.D. IN	Mean OUT	S.D. OUT
1	20.0	7.0	2.00	1.00	0.00	1.00

Preferred orientation of 30. to 40. deg

Locations and Values of Observations

I= 77.0	Y= 77.0	Variable 1= -.386	
I= 7.0	Y= 77.0	Variable 1= -.126	
I= 44.0	Y= 75.0	Variable 1= 1.476	
I= 16.0	Y= 71.0	Variable 1= 1.094	
I= 50.0	Y= 71.0	Variable 1= -.016	
I= 58.0	Y= 66.0	Variable 1= 2.980	
I= 27.0	Y= 63.0	Variable 1= 0.000	
I= 31.0	Y= 61.0	Variable 1= .764	
I= 67.0	Y= 60.0	Variable 1= -.047	
I= 74.0	Y= 55.0	Variable 1= .639	
I= 2.0	Y= 54.0	Variable 1=-1.286	
I= 41.0	Y= 53.0	Variable 1= -.427	
I= 47.0	Y= 49.0	Variable 1= 2.662	
I= 11.0	Y= 47.0	Variable 1=-1.426	
I= 56.0	Y= 43.0	Variable 1= -.262	
I= 28.0	Y= 41.0	Variable 1=-1.013	
I= 63.0	Y= 38.0	Variable 1= .391	
I= 27.0	Y= 36.0	Variable 1= 2.024	
I= 74.0	Y= 31.0	Variable 1=-1.290	
I= 36.0	Y= 30.0	Variable 1= -.539	
I= 40.0	Y= 27.0	Variable 1= .477	
I= 4.0	Y= 25.0	Variable 1= 1.474	
I= 50.0	Y= 21.0	Variable 1= 1.595	
I= 11.0	Y= 20.0	Variable 1=-1.000	
I= 57.0	Y= 16.0	Variable 1= -.762	
I= 20.0	Y= 14.0	Variable 1= -.286	
I= 66.0	Y= 7.0	Variable 1=-1.167	
I= 31.0	Y= 0.0	Variable 1= -.195	
I= 75.0	Y= 3.0	Variable 1= -.079	
I= 39.0	Y= 2.0	Variable 1= 1.017	

Expected number of targets is 1.21

Plotted Frequencies

Probability class	Plot symbol	Frequency
0.0 to 0.0001	.	1122
0.0001 to .1	-	1926
.1 to .2	1	32
.2 to .3	2	40
.3 to .4	3	2
.4 to .5	4	4
.5 to .6	5	3
.6 to .7	6	5
.7 to .8	7	7
.8 to .9	8	31
0.9 to 0.9999	9	0
0.9999 to 1.0	*	0

Figure 2-- Example of FINDER output showing the results for an elliptical target with a preferred orientation. Target and sample locations added.

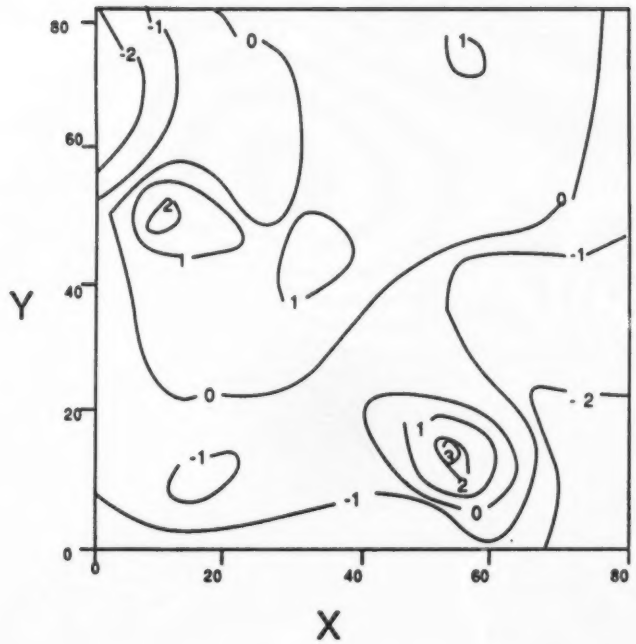
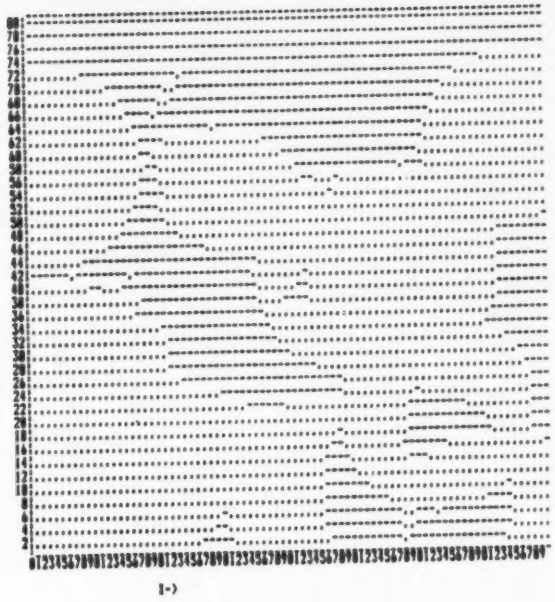


Figure 3A-- Contours of variable 1 listed in Figure 3B. Values represent standard normal deviations with a zero mean.

** FINDER program **

Test of FINDER versus Contoured data

Prior Probability= .0050



Variable Target Sizes, Associated Means and Standard Deviations WithIN & OUTside Target

	Radius	Mean IN	S.D. IN	Mean OUT	S.D. OUT
Var 1	15.0	2.00	1.00	0.00	1.00

Locations and Values of Observations

I=	11.0	Y=	78.0	Variable 1=	-.515
I=	56.0	Y=	75.0	Variable 1=	1.026
I=	74.0	Y=	64.0	Variable 1=	-.189
I=	33.0	Y=	60.0	Variable 1=	-.498
I=	76.0	Y=	60.0	Variable 1=	-.491
I=	3.0	Y=	58.0	Variable 1=	-2.038
I=	6.0	Y=	54.0	Variable 1=	2.010
I=	19.0	Y=	52.0	Variable 1=	1.297
I=	67.0	Y=	50.0	Variable 1=	.056
I=	29.0	Y=	49.0	Variable 1=	-.057
I=	39.0	Y=	44.0	Variable 1=	1.270
I=	50.0	Y=	43.0	Variable 1=	-1.343
I=	50.0	Y=	42.0	Variable 1=	-1.110
I=	2.0	Y=	38.0	Variable 1=	-.486
I=	7.0	Y=	27.0	Variable 1=	-.567
I=	44.0	Y=	23.0	Variable 1=	-.238
I=	63.0	Y=	21.0	Variable 1=	-1.983
I=	24.0	Y=	21.0	Variable 1=	.479
I=	16.0	Y=	13.0	Variable 1=	-1.229
I=	12.0	Y=	13.0	Variable 1=	-.374
I=	24.0	Y=	10.0	Variable 1=	-.482
I=	60.0	Y=	9.0	Variable 1=	3.010
I=	45.0	Y=	8.0	Variable 1=	-1.433
I=	3.0	Y=	3.0	Variable 1=	-1.334
I=	72.0	Y=	3.0	Variable 1=	-2.250

Expected number of targets is .01

Plotted Frequencies

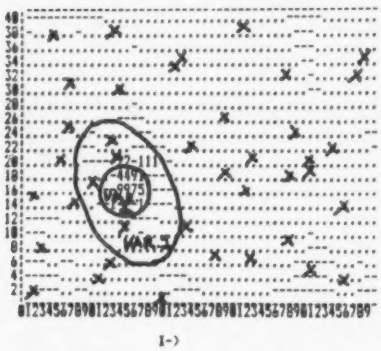
Probability class	Plot symbol	Frequency
0.0 to 0.0001	.	2009
0.0001 to 0.1	-	1191
.1 to .2	1	0
.2 to .3	2	0
.3 to .4	3	0
.4 to .5	4	0
.5 to .6	5	0
.6 to .7	6	0
.7 to .8	7	0
.8 to .9	8	0
.9 to 0.9999	9	0
0.9999 to 1.0	*	0

Figure 3B-- FINDER output from the random normal values of the barren population contoured in Figure 3A.

** FINDER PROGRAM **

Test of Two Variables

Prior Probability= .0010



Variable Target Sizes, Associated Means and Stand. Dev. WithIN & OUTside Target

	Semimajor axis	Semiminor axis	Mean IN	S.D. IN	Mean OUT	S.D. OUT
Var 1	9.7	5.6	2.00	1.00	0.00	1.00
Var 2	2.6	2.5	3.00	1.00	0.00	1.00

Preferred orientation of130. to160. deg

Locations and Values of Observations

X= 32.0	Y= 39.0	Variable 1= -.440	Variable 2= -.370	
X= 14.0	Y= 39.0	Variable 1= .560	Variable 2= .850	
X= 5.0	Y= 37.0	Variable 1= .610	Variable 2= -.720	
X= 23.0	Y= 35.0	Variable 1=-1.710	Variable 2= -.290	
X= 49.0	Y= 35.0	Variable 1= -.370	Variable 2= 1.360	
X= 22.0	Y= 34.0	Variable 1= 1.450	Variable 2= .890	
X= 48.0	Y= 32.0	Variable 1= -.660	Variable 2= .890	
X= 38.0	Y= 32.0	Variable 1= .400	Variable 2= -.160	
X= 7.0	Y= 31.0	Variable 1= -.750	Variable 2= -.150	
X= 14.0	Y= 30.0	Variable 1= -.890	Variable 2= .300	
X= 29.0	Y= 27.0	Variable 1= -.440	Variable 2= -.960	
X= 7.0	Y= 25.0	Variable 1=-1.310	Variable 2= .750	
X= 39.0	Y= 24.0	Variable 1= .310	Variable 2=-2.060	
X= 24.0	Y= 23.0	Variable 1= -.500	Variable 2=-1.020	
X= 13.0	Y= 23.0	Variable 1= 1.900	Variable 2=-1.170	
X= 44.0	Y= 21.0	Variable 1= -.650	Variable 2= -.240	
X= 33.0	Y= 20.0	Variable 1= .100	Variable 2= .620	
X= 6.0	Y= 20.0	Variable 1= -.400	Variable 2= -.700	
X= 41.0	Y= 20.0	Variable 1=-1.240	Variable 2= .190	
X= 14.0	Y= 20.0	Variable 1= 2.600	Variable 2= .080	
X= 38.0	Y= 19.0	Variable 1= -.740	Variable 2=-1.770	
X= 41.0	Y= 19.0	Variable 1= 1.490	Variable 2=-1.510	
X= 29.0	Y= 19.0	Variable 1= -.870	Variable 2= .620	
X= 11.0	Y= 17.0	Variable 1= 3.200	Variable 2= -.460	
X= 32.0	Y= 17.0	Variable 1= -.820	Variable 2= -.570	
X= 2.0	Y= 15.0	Variable 1= -.330	Variable 2= -.610	
X= 46.0	Y= 14.0	Variable 1=-1.230	Variable 2= .330	
X= 6.0	Y= 14.0	Variable 1= -.260	Variable 2= -.510	
X= 15.0	Y= 12.0	Variable 1= 1.670	Variable 2= -.540	
X= 23.0	Y= 11.0	Variable 1=-1.050	Variable 2= .100	
X= 38.0	Y= 9.0	Variable 1= .190	Variable 2=-1.440	
X= 3.0	Y= 8.0	Variable 1= 3.140	Variable 2= .770	
X= 28.0	Y= 7.0	Variable 1= -.970	Variable 2=-1.360	
X= 33.0	Y= 7.0	Variable 1= .200	Variable 2= -.150	
X= 13.0	Y= 6.0	Variable 1=-2.590	Variable 2= -.630	
X= 41.0	Y= 5.0	Variable 1= 1.430	Variable 2= 1.740	
X= 11.0	Y= 4.0	Variable 1= -.670	Variable 2= .080	
X= 46.0	Y= 4.0	Variable 1= .500	Variable 2= -.830	
X= 2.0	Y= 2.0	Variable 1=-1.030	Variable 2= .730	
X= 20.0	Y= 0.0	Variable 1= -.800	Variable 2= -.660	

Expected number of targets is .99

Plotted Frequencies

Probability class	Plot symbol	Frequency
0.0 to 0.0001	-	835
0.0001 to .1	-	149
.1 to .2	1	7
.2 to .3	2	2
.3 to .4	3	0
.4 to .5	4	2
.5 to .6	5	1
.6 to .7	6	0
.7 to .8	7	1
.8 to .9	8	0
.9 to 0.9999	9	3
0.9999 to 1.0	+	0

Figure 4-- FINDER output for two variables. Target and sample locations added.

```
PROGRAM FINDER;          (by D.A.Singer)

CONST nsamples=1000;nvariables=6;
VAR

                                        outunit :text;
        k,l,nvar,line,yplot,xplot,m,ix,sample :integer;
       minimum_or,maximum_or,annulus,jj,mm,ii :integer;
            k2,kkk,iii,nvarp,i,iphi,iss,jb,j :integer;
      dist_cell,kount,skip,istart,count,nsamp :integer;
                                          title :string[80];
                                       ellipses :boolean;
                                      preferred :char;
                                            map :packed array[1..120]of char;
                                      frequency :array[1..12]of integer;
                                  probm,probb,x :array[1..nsamples,1..nvariables]of real;
            a_axis,b_axis,asquare,bsquare,ba :array[1..nvariables]of real;
                                        dim,dib :array[1..100]of real;
                                        mean,sd :array[1..nvariables,1..2]of real;
                              highp,highx,highy :array[1..250]of real;
                                      cell_prob :array[1..120]of real;
        phi,theta,printscale,summ,sumb,prisum :real;
          oneminus_p,expect_number,temp2,temp :real;
     area,prior,degrees,low_orient,high_orient :real;
 upper_y,lower_y,total,horiz_dist,vertical_dist :real;
                  yn,distance,sum,maxasquare :real;


 FUNCTION height(value,meanx,stdv:real):real;
 { Calculates height of normal distribution to estimate probabilities }

  VAR
           stand_score :real;
   BEGIN
       stand_score:=(value-meanx)/stdv;
       IF stand_score > 7.0 THEN stand_score:= 7.0;{Reduce underflow}
       height:=0.3989422*exp(-0.5*stand_score*stand_score);
   END;


PROCEDURE sort(left,right,target,k:integer);
{ Sorts data (Quicksort) on the Y axis of plot so that samples
  are ordered with largest Y first }

   VAR       middle :real;

PROCEDURE swap;

   VAR
          hold :real;      mm :integer;
 BEGIN
  FOR mm:=1 TO k DO
    BEGIN
       hold:=x[ii,mm];
       x[ii,mm]:=x[jj,mm];
       x[jj,mm]:=hold;
```

```
        END
      END;  (swap)

    BEGIN  (sort)
      ii:=left;jj:=right;middle:=x[(left+right)div 2,target];
     REPEAT
       WHILE x[ii,target] >middle  DO ii:=ii+1;
       WHILE middle > x[jj,target] DO jj:=jj-1;
         IF ii <= jj THEN
            BEGIN
              swap; ii:=ii+1;jj:=jj-1;END
    UNTIL ii > jj;
      IF left < jj THEN sort(left,jj,target,k);
      IF ii < right THEN sort(ii,right,target,k);
   END; (sort)



   PROCEDURE setup(VAR outunit:text); { Opens output file }

    BEGIN;
       rewrite(outunit,'resout'); (Opens resout file)
       IF printscale > 0.6 THEN (Adjust for Epson printer)
         BEGIN
           writeln(outunit,chr(15)); (Compressed mode Epson printer)
           writeln(outunit,chr(27),'3');  (Line spacing to 2*Y=X)
           skip:=2;
         END;
       writeln(outunit);writeln(outunit);writeln(outunit);
    END;



  PROCEDURE datain;     { Obtains input }

    VAR        j,k :integer;       e :char;
  BEGIN;
    writeln(' ':15,'** FINDER  PROGRAM **');writeln;
    writeln('Enter title of run');     readln(title);
    write('What is the prior probability of a mineral deposit?-->');
    readln(prior);  oneminus_p:= 1.0 - prior;
    write('How many variables are to be used?-->');   readln(nvar);
    nvarp:= nvar + 2;
    write('Are any of the variables elliptical in shape? (y/n)-->');
    readln(e);
    IF e = 'y' THEN ellipses:=true  ELSE ellipses:=false;
  { If any variables are elliptical in shape then all are assumed to
   be elliptical--however an ellipse can be made nearly circular  }
     IF ellipses THEN
      BEGIN
       FOR k:=1 TO nvar DO   { Input for ellipses }
         BEGIN
          write('Enter semimajor and semiminor axes of variable',k:3,'-->');
           readln(a_axis[k],b_axis[k]);
          write('Enter mean and standard deviation WITHIN ellipse',k:3,'-->');
           readln(mean[k,1],sd[k,1]);
```

```
          write('Enter mean and standard deviation OUTSIDE ellipse',k:3,'-->');
           readln(mean[k,2],sd[k,2]);
         END
       END
      ELSE
       BEGIN
        FOR k:=1 TO nvar DO  ( Input for circles )
         BEGIN
          write('Enter radius of variable',k:3,'-->');    readln(a_axis[k]);
          b_axis[k]:=a_axis[k];
          write('Enter mean and standard deviation WITHIN circle',k:3,'-->');
           readln(mean[k,1],sd[k,1]);
          write('Enter mean and standard deviation OUTSIDE circle',k:3,'-->');
           readln(mean[k,2],sd[k,2]);
         END
      END;
      IF ellipses THEN
       BEGIN
       write('Is there a known preferred orientation to the target? (y/n)-->');
  ( Allows uniform distribution of orientations over restricted range )
       readln(preferred);
       IF preferred='y' THEN
        BEGIN
  write('What are the minimum and maximum orientations? (1-180--90=east)-->');
         readln(low_orient,high_orient);
        END
       END;
     write('Are any variables annular in shape? (y/n)-->');readln(e);
  ( Allows annular shapes with inner ring a circle--will automatically
   generate a new variable and input observations for the inner ring )
     IF e = 'y' THEN
       BEGIN
         ii:=nvar;mm:=1;
             write('Which variable? 1 to ',ii:4,' -->');readln(mm);
             write(' What is the inner radius? -->');readln(temp2);
             nvar:=nvar +1;
             mean[nvar,1]:= mean[mm,2];   sd[nvar,1]:=sd[mm,2];
             mean[nvar,2]:= mean[mm,1];   sd[nvar,2]:=sd[mm,1];
             a_axis[nvar]:= temp2;          b_axis[nvar]:=0.99 * a_axis[nvar];
             annulus:= mm;
        END;
    write('Enter number of cells to be plotted in X and Y direction-->');
 ( Maximum X and Y cell locations to be plotted--all maps have lower
   left locations of 1,1--X is limited to not more than 120 cells,
   Y is unlimited )
     readln(xplot,yplot);
     write('Enter printer scale, 0.3 or 0.6 for standard,1.0 for Epson-->');
     readln(printscale);
    write('Enter number of observations-->');  readln(nsamp);
writeln('Enter X and Y Locations and Responses for each variable--1 set/line');
( Input observations X,Y,Var1,Var2,...Varn--one location per line--X,Y same
  scale as map cells )
    FOR k:= 1 TO nsamp DO
      BEGIN
       write('Observation',k:3,'-->');
       read(x[k,1],x[k,2]);
```

```
        FOR j:= 3 TO nvarp DO
          BEGIN
            read(x[k,j]);
          END;
        readln;
 { Makes new variable for inner ring of annulus }
        IF annulus > 0 THEN   x[k,nvarp + 1]:=x[k,annulus+2];
       END;
      IF annulus > 0 THEN nvarp:= nvarp + 1;
 END;


  PROCEDURE maxa;

    VAR        i :integer;
     BEGIN;
        area:=3000.0;
        maxasquare:=0.0;
        IF annulus > 0 THEN temp2:=a_axis[annulus]*b_axis[annulus] - a_axis[nvar]*a_axis[nvar];
       FOR i:= 1 TO nvar DO
         BEGIN
            asquare[i]:=a_axis[i]*a_axis[i];
            bsquare[i]:=b_axis[i]*b_axis[i];
            ba[i]:=b_axis[i]/a_axis[i];
            temp:=a_axis[i]*b_axis[i];
            IF asquare[i] > maxasquare THEN maxasquare:=asquare[i];
            IF temp < area THEN area:=temp;
         END;
{ Determines area of smallest variable to estimate number of deposits }
       IF temp2 < area THEN area:= temp2;
       IF (annulus > 0) and (nvar = 2) THEN area:= temp2;
       area:=area*3.14159;
     END;



PROCEDURE fillprob; (Calculates probabilities for each sample & variable )

    VAR       k,l :integer;
  BEGIN
   FOR l:=1 TO nsamp DO
     BEGIN
       FOR k:=1 TO nvar DO
         BEGIN
           proba[l,k]:=height(x[l,k+2],mean[k,1],sd[k,1]);
           probb[l,k]:=height(x[l,k+2],mean[k,2],sd[k,2]);
         END
     END;
   END;


PROCEDURE zero;  (Initalizes arrays )

  VAR           k :integer;
  BEGIN
    FOR k:=1 TO 12 DO
```

```
        BEGIN
          frequency[k]:=0;
        END;
      FOR k:=1 TO 180 DO
        BEGIN
          dia[k]:=0.0;
          dib[k]:=0.0;
        END;
      FOR k:=1 TO 120 DO
        BEGIN
          map[k]:=' ';
        END;
    END; {zero}


PROCEDURE altern(k:integer);
{ Used to fill hit arrays where orientations rap around 180 degrees }

  VAR       l :integer;
  BEGIN
   FOR l:=1 TO iphi DO
     BEGIN
       j:=i;
       IF i > 180 THEN j:=i-180;
       IF (dia[j]> 1.0E-15) and (dib[j]> 1.0E-15) THEN
          BEGIN { Prevents underflows}
            dia[j]:=dia[j]*proba[ma,k];
            dib[j]:=dib[j]*probb[ma,k];
          END
       ELSE
       IF dia[j] = 0.0 THEN
          BEGIN
            dia[j]:=proba[ma,k];
            dib[j]:=probb[ma,k];
          END;
       i:= i + 1;
     END;
   minimum_or:=1;maximum_or:=180;
  END; {altern}


PROCEDURE calc2;  { Calculates probabilities for elliptical variables }

 VAR       jb,k :integer;    two_radians :real;
  BEGIN
   two_radians:=57.295778 * 2.0;
   FOR k:=1 TO nvar DO
     BEGIN
       IF distance <= asquare[k] THEN
         BEGIN
          IF distance <= bsquare[k] THEN    { Inner circle }
             BEGIN
                i:=1;
                iphi:=180;
             END
          ELSE
```

```
            BEGIN
              IF vertical_dist = 0.0 THEN theta:=90.0
                 ELSE (theta is angle in degrees between cell & sample loc.)
                      theta:=arctan(horiz_dist/vertical_dist)*57.29578;
              phi:=arctan(ba[k]*sqrt((asquare[k]-distance)/(distance-bsquare[k])));
              iphi:=trunc(phi*two_radians +0.4999); (iphi is number of hit degrees)
              i:=trunc(theta-phi*57.29578 + 0.4999);(i is 1st hit or. in deg)
              IF i <= 0 THEN i:=i + 360;
              IF i > 180 THEN i:=i - 180;
            END;
          IF i + iphi > 181 THEN altern(k)
            ELSE
              BEGIN
                iss:=iphi + i - 1;
                FOR jb:= i TO iss DO (fills hit orientation arrays dia,dib)
                  BEGIN
                   IF (dia[jb]> 1.0E-15) and (dib[jb]> 1.0E-15) THEN
                      BEGIN
                        dia[jb]:=dia[jb]*proba[aa,k];
                        dib[jb]:=dib[jb]*probb[aa,k];
                      END
                   ELSE
                   IF dia[jb] = 0.0 THEN
                      BEGIN ( First time orientation affected )
                        dia[jb]:=proba[aa,k];
                        dib[jb]:=probb[aa,k];
                      END;
                   END;
                IF i < minimum_or THEN minimum_or:=i;
                IF iss > maximum_or THEN maximum_or:= iss;
              END
          END;
        END;
      END; (calc2)


PROCEDURE calc3;
(Calculates probabilities of in target (suma) & out of target (sumb)
 for circular VARiables )

    VAR     k :integer;
  BEGIN
     FOR k:= 1 TO nvar DO
         IF distance <= asquare[k] THEN
            BEGIN
               suma:=suma*proba[aa,k];
               sumb:=sumb*probb[aa,k];
            END;
  END;  (calc3)


PROCEDURE calc;

  VAR      arc,sample,l,ix :integer;

  BEGIN
```

```
    FOR ix:=1 TO xplot DO    (Cycle through cells on a line)
        BEGIN
            kkk:=0;
            maximum_or:=1;minimum_or:=180; (Maximum & minimum orientation)
            suma:=1.0;sumb:=1.0;
            FOR sample:= istart TO count DO (Cycle through samples)
               BEGIN
                  m:=sample;
                  horiz_dist:=x[m,1] - ix;  (Horizontal distance cell & sample)
                  vertical_dist:=x[m,2] - yn;  (Vertical distance cell & sample)
                  distance:= horiz_dist*horiz_dist + vertical_dist*vertical_dist;
  { Determines if cell is within range of any sample }
                  IF distance <= maxasquare THEN IF ellipses THEN
                      BEGIN     { Calculates probabilities for ellipse }
                          kkk:=1;
                          calc2;
                      END
                  ELSE calc3;    { Calculates probabilities for circle }
               END;
            IF kkk > 0  THEN
               BEGIN   { Ellipse }
                  arc:=0;
                  suma:=1.0E-34;
                  IF preferred='y' THEN
                     BEGIN      { Restricted range of orientations }
                       minimum_or:=trunc(low_orient);  maximum_or:=trunc(high_orient);
                     END;
                 FOR l:= minimum_or TO maximum_or DO
                     BEGIN
                       IF dia[l] > 0.0 THEN
                          BEGIN   { Bayes update for elliptical variables }
                            prisum:=prior*dia[l];
                            suma:=suma + prisum/(prisum + oneminus_p*dib[l]);
                            arc:=arc +1;
                          END;
                       dia[l]:=0.0;
                       dib[l]:=0.0;
                     END;
  { Account for partial information where only some orientations affected }
                  cell_prob[ix]:=(suma+prior*(degrees-arc)) /degrees;
               END;
           IF(suma>0.0) and (sumb>0.0) THEN
                BEGIN
                  IF kkk < 1 THEN
                     BEGIN     { Bayes update for circular variables }
                        prisum:=prior*suma;
                        cell_prob[ix]:=prisum/(prisum + oneminus_p*sumb);
                     END
                END;
           iii:=trunc(cell_prob[ix]*10.002 + 1.999); (Converts prob. to integer)
           frequency[iii]:=frequency[iii] + 1;
           total:=total + cell_prob[ix];
                CASE iii of     { Assigns plot symbol to probability }
                      1:map[ix]:='.';
                      2:map[ix]:='-';
                      3:map[ix]:='I';
```

```
                    4:map[ix]:='2';
                    5:map[ix]:='3';
                    6:map[ix]:='4';
                    7:map[ix]:='5';
                    8:map[ix]:='6';
                    9:map[ix]:='7';
                   10:map[ix]:='8';
                   11:map[ix]:='9';
                   12:map[ix]:='*';
              END (case)
        END; (for)
    END;   ( calc )


PROCEDURE number_deposits;

  VAR                    lower,upper :integer;
        tempy,xdif,ydif,dist,tempo :real;

 BEGIN
  lower:=1;
  expect_number:=0.0;
  kount:=kount + 1;
  highp[kount]:=0.0;
  IF kount = 2 THEN expect_number:= highp[1]
   ELSE
    BEGIN
     WHILE lower <= kount DO
       BEGIN
         tempo:=highp[lower];
         FOR upper:= lower+1 TO kount-1 DO
           BEGIN
              xdif:= highx[lower] - highx[upper];
              ydif:= highy[lower] - highy[upper];
              dist:=sqrt(xdif * xdif + ydif * ydif);
              IF dist <= dist_cell THEN (cells within r each other)
                BEGIN
                  tempy:= highp[upper];
                  highp[upper]:=0.0;
                  highx[upper]:=highx[lower];
                  highy[upper]:=highy[lower];
                  IF tempy > tempo THEN
                     BEGIN
                        tempo:= tempy; (use only highest prob)
                     END  (if)
                END  (if dist )
           END; (for)
         expect_number:= expect_number + tempo;
         lower:= lower + 1;
       END (while)
    END  (else)
END;  (number_deposits)


 PROCEDURE highprob;
```

```
        VAR       tempp,tempx :real;
            istart,cellcount :integer;

    BEGIN
      cellcount:=1;
      REPEAT
        IF cell_prob[cellcount] > prior THEN
           BEGIN
             istart:= cellcount;
             tempp:=cell_prob[cellcount];
             tempx:= cellcount;
             FOR cellcount:=istart TO cellcount + dist_cell DO
                BEGIN
                  IF cell_prob[cellcount] > tempp THEN
                      BEGIN
                         tempp:= cell_prob[cellcount];
                         tempx:= cellcount;
                      END  { if }
                END; { for }
             kount:= kount + 1;
             highp[kount]:= tempp;
             highx[kount]:= tempx;
             highy[kount]:= yn;
           END  {if }
        ELSE
           cellcount:= cellcount + 1;
      UNTIL cellcount > xplot;
   END;  { highprob }


  PROCEDURE summary(VAR outunit:text);      { Writes output file }

       VAR        k,l :integer;   b1,b2 :real;

   PROCEDURE blank;            { Writes 4 blank lines }

        VAR        k :integer;
       BEGIN  {blank}
          writeln(outunit);
          FOR k:=1 TO 4 DO
               BEGIN
                  writeln(outunit);
               END
       END;  {blank}

 BEGIN     {summary}
  blank;
  write(outunit,' ':1,'Variable Target Sizes, Associated Means and');
  writeln(outunit,' Stand. Dev. WithIN & OUTside Target');
  writeln(outunit);writeln(outunit);
  IF ellipses THEN
    BEGIN
     write(outunit,' ':10,'Semimajor axis',' ':5,'Semiminor axis');
     write(outunit,' ':5,'Mean IN',' ':5,'S.D. IN',' ':5,'Mean OUT',' ':5);
     writeln(outunit,'S.D. OUT');
    END
```

```
      ELSE
       BEGIN
        write(outunit,' ':15,'Radius',' ':8,'Mean IN',' ':5,'S.D. IN');
        writeln(outunit,' ':5,'Mean OUT',' ':5,'S.D. OUT');
       END;
     FOR k:=1 TO nvar DO
       IF ellipses THEN
         BEGIN
          write(outunit,'Var',k:3,' ':8,a_axis[k]:6:1,' ':13,b_axis[k]:6:1,' ':8);
          write(outunit,mean[k,1]:7:2,' ':5,sd[k,1]:7:2,' ':6,mean[k,2]:7:2);
          writeln(outunit,' ':5,sd[k,2]:7:2);
         END
       ELSE
         BEGIN
          write(outunit,'Var',k:3,' ':8,a_axis[k]:6:1,' ':8,mean[k,1]:7:2,' ':5);
          writeln(outunit,sd[k,1]:7:2,' ':5,mean[k,2]:7:2,' ':5,sd[k,2]:7:2);
         END;
    blank;
    IF preferred='y' THEN
  writeln(outunit,'Preferred orientation of',low_orient:4:0,' to',high_orient:4:0,' deg');
    writeln(outunit);
    writeln(outunit,' ':10,'Locations and Values of Observations');
    writeln(outunit);
    FOR k:=1 TO nsamp DO
        BEGIN
          write(outunit,'X=',x[k,1]:6:1,'   Y=',x[k,2]:6:1);
          FOR l:=3 TO nvarp DO
            BEGIN
              ii:=l-2;
               write(outunit,'  Variable',ii:2,'=',x[k,l]:6:3);
            END;
          writeln(outunit);
        END;
    blank;
    writeln(outunit,' ':15,'Expected number of targets is',expect_number:6:2);
    blank; writeln(outunit,' ':10,'Plotted Frequencies');writeln(outunit);
    writeln(outunit,'Probability class  Plot symbol  Frequency');
    writeln(outunit,'   0.0 to 0.0001',' ':6,'.',' ':10,frequency[1]:6);
    writeln(outunit,'0.0001 to  .1',' ':9,'-',' ':10,frequency[2]:6);
    b1:=0.0; b2:=0.1;
    FOR k:=1 TO 8 DO
       BEGIN
         b1:=b1 + 0.1; b2:=b2 + 0.1;
         write(outunit,'   ',b1:3:1,' to ',b2:3:1,' ':9,k:1,' ':10);
         writeln(outunit,frequency[k+2]:6);
       END;
   writeln(outunit,'   0.9 to 0.9999',' ':6,'9',' ':10,frequency[11]:6);
   writeln(outunit,'0.9999 to 1.0',' ':9,'*',' ':10,frequency[12]:6);
   blank;
 END;   ( summary )


PROCEDURE plot(VAR outunit:text);    ( Calls calc and writes map )

  VAR          test :boolean;    line,k,l,yline :integer;
             xlabel :string[10];
```

```
BEGIN;
 xlabel:='0123456789';
 kount:=0;
 dist_cell:= trunc(2.0 * (sqrt(maxasquare) + 0.499));
 yline:=trunc(yplot * printscale + 0.499);   (Printer scale adjustment)
 FOR line:=1 TO yline DO    ( Cycle through lines (Y direction) )
    BEGIN
       yn:=(yline - (line - 1))/printscale;
       m:=istart - 1; count:= istart;
( upper_y & lower_y are used in window of largest variable around each line )
       upper_y:=yn + maxasquare;     lower_y:=yn - maxasquare;
       test:=false;
       FOR k:= 1 TO xplot DO     ( Default line )
           BEGIN
              map[k]:='-';
              cell_prob[k]:=prior;
           END;
       k2:=trunc(yn + 0.4999);
(If Epson printer then)
( Skips odd numbered lines to make readable output and save time )
       IF k2 mod skip=0 THEN
         BEGIN
            REPEAT  (Cycle through samples to find any that could affect line)
               m:=m + 1;
               IF x[m,2] >= upper_y THEN
                  istart:=m
                ELSE
                   BEGIN
                     IF x[m,2] < lower_y THEN
                        BEGIN   ( Sample m is below limit for line--stop cycle)
                           count:=m - 1;
                           m:=nsamp;
                        END
                     ELSE
                        BEGIN      ( At least one sample could affect line )
                           count:=m;
                           test:=true;
                        END
                   END;
            UNTIL m=nsamp;
            IF test THEN  calc; (Calculates probabilities & fills output line)
            writeln(outunit,' ':5,k2:5,':',map);    ( Writes one line of map )
            highprob;
         END
       ELSE
            writeln(outunit,' ':10,':');  ( Writes blanks for odd numb.lines)
      END;
       FOR l:=1 TO xplot DO
        BEGIN                 ( Writes line on bottom of map )
          map[l]:='-';
        END;
       writeln(outunit,' ':11,map);
       k:=xplot div 10;
       write(outunit,' ':10);
         FOR l:= 1 TO k DO
             BEGIN;            ( Writes X scale on bottom of map )
```

```
            write(outunit,xlabel);
        END;
    IF printscale > 0.4 THEN (Only for Epson printer)
    writeln(outunit,chr(27),'0');(Changes printer to normal line space)
    writeln(outunit);writeln(outunit,' ':30,'X->');
    number_deposits;
    summary(outunit);     ( Summarizes results )
 END;   ( plot )


BEGIN;    ( Main Program )

  istart:=1;  degrees:=180.0;  preferred:='n';total:=0.0;
  temp2:=3000.0;  annulus:=0;  skip:=1;
  datain;    ( Gets input )
  IF preferred='y'  THEN degrees:=high_orient - low_orient +1;
  sort(1,nsamp,2,nvarp);
  zero;
  maxa;
  writeln('Now calculating');
  setup(outunit);
  fillprob;writeln(outunit,' ':35,'** FINDER PROGRAM **');
  writeln(outunit);writeln(outunit);
  writeln(outunit,' ':20,title);writeln(outunit);writeln(outunit);
  writeln(outunit);writeln(outunit,' ':20,'Prior Probability= ',prior:6:4);
  writeln(outunit);writeln(outunit);
  FOR l:=1 TO xplot DO
    BEGIN
      map[l]:='-';
    END;
  writeln(outunit);writeln(outunit,' ':11,map);  ( Writes line top of map )
  plot(outunit);        ( Calculates and plots map and summary )
  writeln(chr(7));      ( Sounds bell at end of calculations )
  writeln('Finished');
END.  ( FINDER )
```